Privacy Strategies in Smart Metering

Tobias Klenze
Supervisor: Benjamin Hof
Seminar Innovative Internettechnologien und Mobilkommunikation
Chair for Network Architectures and Services
Department of Informatics, Technische Universität München
Email: klenzet@in.tum.de

ABSTRACT

This paper discusses the threat to privacy that smart meters pose and offers solutions that protect user privacy without diminishing the provider's capabilities to use the data for legitimate purposes. Two use cases are considered: collection of aggregated real-time data by the provider for optimization, and monthly billing of customers. For the first, we present various approaches before proposing our own strategy for an aggregation protocol. For the second, we outline a billing protocol that preserves user privacy.

Keywords

Smart metering, Privacy, Data aggregation, Anonymization, Security, Homomorphic encryption.

1. INTRODUCTION

Smart meters currently undergo a deployment in many countries. While their advocates promise benefits to users and the grid infrastructure, the privacy implications of high-frequency data collected by smart meters has received little attention by the industry.

According to goals set by the European commission, 80 percent of alls EU households shall be equipped with smart meters by 2020 [1]. Advocates cite different reasons supporting wide-spread smart meter deployment. Automated billing eliminates the need for periodic inspections of meters by utility providers. It is also anticipated that integration of smart meters into a smart grid will facilitate fraud detection. Furthermore, electricity providers are interested in smart meters, because they enable complex tariffs, charging the customer more in peak consumption hours and less during hours with an abundancy of electricity. By combining smart meters with "smart appliances" that are sensitive to real-time electricity prices, it is possible to reduce the consumption during peak hours. For example, electric cars may charge up during the night, instead of busy hours during the day, thus reducing the capability requirements of the grid during peak hours, ultimately leading to fewer power plants being required. Awareness of current consumption on the user side may also lead to an overall reduction: studies show that 3-5% less energy is consumed when real-time meter readings are provided to consumers [8]. Finally, smart meter advocates state that real-time usage information will lead to a higher efficiency in managing grids and demand forecasting, facilitating decisions such as whether to start up a power plant and detecting leaks in pipelines as early as possible [7].

Whether or not these expectations of the capabilities of a "smart grid" (of which smart meters are a part) are justified, is beyond the scope of this paper. There have already been over three billion Euros of smart grid investment in the EU.[2] The fact of increasing deployment is reason enough to examine the implications of smart meters for consumers, particularly for their privacy. Current smart meters output usage data at a high-frequency, for example every 15 minutes, and report this usage to utility or grid providers. This is in contrast to old electromechanical meters, for which the provider only has low-frequency data, for instance on a monthly basis. As we will present in the next section, the higher frequency in usage reports to the provider has a drastic impact on the user's privacy.

In some countries, efficient energy usage and user privacy are debated as a dichotomy. Either the preference is given towards the benefits of smart meters and privacy detriments accepted as a necessary consequence, or the opposite holds, and smart meters are rejected altogether [7]. We will present protocols that protect the user's privacy while providing a high level of security.

Satisfying privacy and security at the same time is a difficult task, as the use case of billing with time-dependent tariffs illustrates. In this use case, monthly bills should be calculated from high-frequency usage data. On the one hand, the utility has an interest in computing the bill itself, so as to be secure against manipulation by the consumer. On the other hand, it must not gain access to fine-grained usage data for privacy reasons. Protocols that we are going to present in this paper use cryptographic mechanisms to ensure both privacy and security.

When designing a secure and privacy-preserving smart meter system, one also has to take device tampering into account. Attacks on smart meters are not merely theoretical. A 2011 study cites an FBI report, stating that as much as 10% of smart meters in one US state had been tampered with [7].

In this paper, we will present approaches that deal with privacy problems introduced by smart meters. The goal is to give an overview over the set of problems, while restricting the solutions presented to those that illustrate the different strategies in preserving the privacy of smart meter users.

We will start by presenting privacy problems in smart me-

doi: 10.2313/NET-2014-08-1_16

ters and then discuss at strategies that mitigate these issues while retaining the desired features of smart meters that smart meter advocates anticipate.

2. PROBLEM STATEMENT

We distinguish two different types of analysis that have implications for user privacy. The first is a statistical analysis that makes it possible to identify household appliances, such as a TV, a shower or a toaster. The second one is aimed at providing more abstract information, such as whether a person suffers from insomnia, or what religious group a resident belongs to. For the second type, it might be necessary to first identify individual appliances. Not all abstract questions about a household require such analysis however.

2.1 Identifying appliances

It has been shown over the last years, that smart meter data of sufficient quality can be used to identify single appliances with great certainty using statistics and machine learning. Existing approaches use high-frequency usage data, mainly of electricity consumption, to identify a number of household devices, such as televisions, washing machines, computers, refrigerators and even low-powered devices such as alarm clocks [11].

Figure 1 shows an example, where power consumption has been recorded over the period of one day. Alongside the consumption is annotated which appliances and which abstract information the authors of [9] have been able to identify using statistical methods and machine learning.

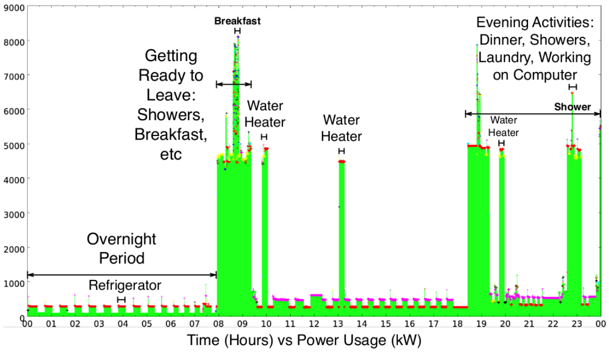

Figure 1: A sample identification of appliances using signatures. Source: [9]

2.2 Profiling on behavioral patterns

As the graph above demonstrates, using appliance signatures, it is possible to gain more abstract insights and do user profiling. For example, data on individual appliances can be utilized to answer questions such as: "Is the resident at home?" Imagining a utility provider asking this question about their employees on sick leaves, one can see why this data is sensitive and worthy of protection. After individual appliances have been identified, it is possible to answer these kinds of abstract questions [9].

It is difficult to tell which kind of data enables to such profiling. Some conclusions require accurate, high-frequency data, while others can be made with low-frequency and less accurate information. Profiling can not be completely prevented, but only limited. Abstract questions of the form "How many residents does a household have" or "Did all the residents go on vacation" can still be answered, even after applying the methods presented in this paper. For these questions, an analysis of very low-frequency data, maybe even as low-frequency as the monthly bill is sufficient. Evaluating how much profiling the data provided in a certain protocol permits is thus difficult. The more accurate, high-frequency data is available to the adversary, the easier profiling becomes. We will therefore aim to minimize accurate, high-frequency data availability to the utility and third parties.

3. SOLVING THE PRIVACY PROBLEMS

We will proceed to present solutions to the problems outlined in the last section. Two different use cases for high-frequency data will be distinguished and discussed separately.

3.1 Applications for smart meter data

In recent years, a lot of papers have discussed smart meter privacy problems. They are very diverse in the problem statement and the proposed solutions. For instance, some research how smart meter data sharing with third parties is possible in a privacy-preserving manner, and presuppose that the utility has access to sensitive information [6, 13]. This is contrasted by most papers which consider the relationship between the end user and the utility provider (and potentially a smart grid operator).

Leaving special cases aside, there turn out to be two common use cases for high-frequency smart meters data that have important privacy implications. The first is real-time data sharing with the utility and the grid provider. This data is used for management of the grids and plants, to have accurate and live information on utility usage, for demand forecasting, to detect problems and leaks. It has been noted that this use case is difficult to define, since there is no detailed specification on which data utility providers and grid operators require to achieve these goals [4]. Some providers try to collect as much data as possible, which is contrary to one of the most important principles in privacy design strategies, that data collection should be minimized [5]. Whether consumption data from individual households is actually necessary to achieve the goals outlined above, or whether the accuracy and high frequency is required, is doubtful. We will present approaches that contest these assumptions in order to allow for protocols that protect user privacy.

The second use case of high-frequency metering data is billing with non-constant tariffs. We will outline a scheme that allows for sophisticated tariffs, while preserving user privacy and keeping complex computations outside the smart meter's limited hardware.

The last point is an important consideration. Smart meters are usually low-end systems both in terms of computational power and bandwidth [7]. Solutions which require computationally expensive operations or different hardware than what currently-produced smart meters offer, are less likely to be deployed by providers. In addition, communication overhead should be kept to a minimum. It is due to these reasons that protocols for smart meters are usually de-signed to have minimal computations inside the trusted and tamper-resistant device itself. Instead, they only use the smart meter to do basic operations and outsource computationally expensive operations to other, perhaps untrusted devices.

3.2 Real-time data sharing

This section presents and discusses strategies for providers to capture real-time usage information without diminishing the user's privacy. We will first present two approaches that we regard not to be a sufficient solution to the problems outlined in the last section and then take a closer look at a more sophisticated protocol, which provides better protection of user privacy. Following our critique of this approach, we present our own modification of the protocol which aims at protecting users in a better way against maliciously acting smart meters.

3.2.1 Preprocessing

Preprocessing data is a strategy to hinder or at least to impede inferences that usage data permits. The idea is to obfuscate some of the information, making detailed analysis harder, without having an effect on the use of aggregated data. By preprocessing real-time usage data in certain ways, it is harder to identify appliances and thus it is also more difficult to come to certain abstract conclusions. For example, the data might still reveal that a person is in a household, but not whether she is working, or watching TV.

In [11], Reinhardt, Englert, and Christin discuss two different ways of data preprocessing: quantization and down-sampling. The first is restricting the possible values of reported power consumption to a multiple of a certain factor. The latter reduces the frequency with which readings are reported. For instance, one might employ both techniques and record usage only as a multiple of $q = 100$ *watts* (instead of a higher precision). Then, instead of reporting this consumption at a very high frequency, for instance every second, it is reported only every $t = 900$ *s*.

The authors evaluate both methods on their own and in combination. In their model, quantization is effective, whereas down-sampling has a smaller impact on appliance detection rate. Before applying any kind of preprocessing, their machine learning system is able to identify more than 90 percent of appliances in a sample. Applying quantization with $q = 180$ *watts* decreases accuracy to 58 percent, while applying down-sampling only decreases it to 74 percent with $t = 400$ *s*. Combining both techniques with said values still leaves the machine learning system able to identify 38 percent of all appliances.[1]

While making it more difficult for adversaries to draw conclusions given the usage data of a household, preprocessing is far from making it impossible altogether. Even with a detection rate of only 38 percent, there remain inferences that can be made from the preprocessed data, that problematic in terms of user privacy.

3.2.2 Multi-party protocols

Since preprocessing data is not sufficient to fully solve privacy problems with real-time smart meter data, we turn to a different approach. The assumption behind it is that providers do not require real-time data from each household. Instead, they can rely on aggregated data from multiple households. By obscuring the individual contributions of households to the aggregated value, their privacy is upheld.

A naive implementation of this concept is to employ a trusted third party (TTP), that aggregates each meter's readings and passes on the total to the provider. This proposal only shifts the privacy problems to the TTP. There are a number of authors who propose aggregation protocols without a TTP [7, 3, 4, 14]. For instance, [3] addresses the privacy problem by distributing data over different providers. The smart meter divides its consumption data into K different shares (which summed up, are the actual consumption), each of which is sent to a different provider. The idea is that a provider cannot infer information about an individual household with only a share of the actual consumption. However, each provider can aggregate the data and share that with other providers, thus enabling them to use high-frequency data without gaining insights on any individual household.

This approach is more promising than using a TTP, but it has a severe problem: Individual privacy is lost when an adversary learns the data shares sent to all K different providers. This could be due to collusions by the providers, but a perhaps more realistic scenario is when they are forced to hand over information about individual shares to a government agency. We therefore conclude that this strategy does not provide adequate protection in these adversarial models.

3.2.3 "No-Leakage" aggregation protocol

We will now present an aggregation protocol (called the "No-Leakage Protocol") that was proposed by Garcia and Jacobs [4]. We will then discuss drawbacks of the underlying privacy strategy and offer a modified version of the protocol in the next section.

The protocol gives the provider aggregated real-time consumption data of multiple households without revealing their individual shares. It uses homomorphic encryption in an asymmetric cryptography setting.

Homomorphic Encryption. An encryption scheme is homomorphic if it allows for sensible operations being carried out on ciphertext. For example, this could be an operation $\oplus$ on two encrypted values $enc(m_1)$, $enc(m_2)$, such that $enc(m_1) \oplus enc(m_2) = enc(m_1 + m_2)$. This encryption scheme is then called additively homomorphic.

A common additively homomorphic scheme is Paillier's cryptosystem, where:

$$dec(enc(m_1) \cdot enc(m_2) \mod n^2) = m_1 + m_2 \mod n$$

It is possible to multiply the encrypted value by a constant:

$$dec(enc(m)^k \mod n^2) = m \cdot k \mod n$$

[1] It should be noted that [11, 9] started with a frequency of $1Hz$, which is higher than current smart meter's output frequency.

Setting and Infrastructure. The setting which we assume to work on consists of a local distribution station S and N households, each with a smart meter M_i connected to S. The goal is to provide near real-time usage information in aggregated form to the distribution station without revealing any individual contribution to the total consumption. The local station has a direct data connection to every household's smart meter. All data connection lines are assumed to secure the integrity of messages exchanged between two directly adjacent parties.

In the setup phase, each party (all M_i and S) creates a public/private key pair and we assume that there is a mechanism which distributes the public keys to all parties (e.g. by using a PKI).[2] It is assumed that the encryption scheme used is additively homomorphic, with $\oplus$ being multiplication (as in the example above).

The protocol may require a minimum number of participating smart meters. Since the security of the protocol is weakened with too few participants, the smart meters should not execute the data exchange if there are fewer than the minimum number of smart meters.

Protocol flow. After each measurement interval (for example, every 15 minutes), smart meters report on their household's consumption. Instead of sending their usage data directly to the local station S, they prepare N shares that are each sent to a different of the $N-1$ smart meters connected to the local station, except for one share, which remains at the meter. The shares are each encrypted using the recipient smart meter's public key (so that the local station, which has to relay the messages, does not learn the shares in plaintext). More formally, each smart meter M_i computes:

- Its own consumption m_i.
- Random numbers $a_{i1}, a_{i2}, \ldots a_{i(N-1)}$, and $a_{iN} \equiv m_i - \sum_{j=1}^{N-1} a_{ij} \ mod\ n$ (using a sufficiently large n).
- Encrypted values $y_{ij} = enc_{pk_j}(a_{ij})$ for each $1 \leq j \leq N$, $j \neq i$, where pk_j is j's public key exchanged during setup. a_{ii} is not encrypted and remains at the meter.

All the smart meters then send their y_{ij} values to S. S then computes for each j:

$$s_j := \prod_{i \neq j} y_{ij} = \prod_{i \neq j} enc_{pk_j}(a_{ij}) = enc_{pk_j}(\sum_{i \neq j} a_{ij})$$

Note that this equation holds due to the additive homomorphic encryption properties of the underlying cryptosystem.

S sends s_j to M_j, for all j.

M_j decrypts s_j using his private key sk_j, yielding $\sum_{i \neq j} a_{ij}$. To this, M_j adds his own a_{jj}. M_j finally sends this value to S:

$$r_j := \sum_{i \neq j} a_{ij} + a_{jj} \ mod\ n$$

The addition of a_{jj} is essential, since otherwise S could exploit M_j's willingness to decrypt any ciphertext. If S probed M_j with individual shares (y_{ij}) instead of aggregated shares (s_j), M_j would return the plaintext shares (a_{ij}) to S. To avoid this, the value of r_j is obscured by the random value a_{jj}. As a final step of the protocol, S sums up all reported values:

$$total := \sum_j r_j = \sum_j \sum_{i \neq j} a_{ij} + a_{jj} \ mod\ n = \sum_i \sum_j a_{ji} \ mod\ n$$

This equals the total consumption of all households.

Proofs. Alongside the description of this protocol, [4] includes a proof of correctness and a proof of a "no leakage" property. For the former, it is merely noted that each a_{ij} is sent exactly once in a r_j message to S (by meter j) and that the sum of all shares a_{ij} over i and j is equal to the total consumption.

The "no leakage" property states that even if all but two meters are corrupted and act to reveal the uncorrupted meter's usage, no information about the latter meters' readings is leaked. This is an important result, since it means it is infeasible for colluding parties (e.g. S and a number of meters) to learn the consumption of other meters.

To accurately define this property, a "no leakage game" is defined. The proof relies on the underlying encryption scheme being IND-CPA secure (which is a formal definition of security in context of encryption). It shows that an adversary who wins the "no leakage game" can also win the IND-CPA-Game of the encryption scheme.[3] Presenting this proof is beyond the scope of this paper. Instead, we will discuss disadvantageous design choices of the protocol in the next paragraph and present a variation of the protocol addressing these problems in the next section.

Discussion. The "No-Leakage Protocol" is a protocol between a local station (operated by the provider) and individual household meters. It gives smart meters more control over the process of aggregation and limits the information that providers can learn. It therefore fits well the author's paradigm "Power to the Meter" which could be interpreted to be contrary to the status quo, which gives "Power to the Provider".

[2] The establishment of trust is an important challenge, but cannot be resolved in this paper. For our purposes, we assume that each party receives the correct public key of each other party.

[3] The paper does not discuss the underlying encryption in detail, but mentions Paillier's cryptosystem as being IND-CPA secure.

Taking power from the provider makes it more difficult for them to learn anything about a household's real-time consumption, but depending on the adversary model, it does not make it impossible. In the "Power to the Meter" setting proposed by the protocol's authors, meters are assumed to be trustworthy. In a scenario where the utility or grid operator is assumed to be the adversary trying to extract real-time usage data, this is a dubious assumption. Providers could influence the trusted computing base of smart meters, making all smart meters corrupted and thus violating the premise required for the security proof.

If all meters are corrupted, a naive way for them to leak usage information while following the protocol is to let each smart meter i set $a_{ij} = 0$ for all $i \neq j$ and $a_{ii} = m_i$. Then the actual consumption is revealed to the station S. To counter this attack, one might allow the user (i.e. household residents) to monitor the connection between their smart meter and S. The users would be able to detect such a simple attack. However, there are more sophisticated attacks, where detection is not possible. For example, a_{ij} for all i and j s.t. $i \neq j$ might be chosen from a pseudorandom sequence that is available to all the smart meters and the local station S. Each M_i uses these pseudorandom values, and sets a_{ii} in a way such that over all j, a_{ij} add up to m_i (as the protocol demands). To learn the consumption by meter j, S has to subtract the sum of all a_{ij} over i, s.t. $i \neq j$ from r_j (these a_{ij} values are known to S, since they belong to the pseudorandom sequence that S and all meters share). The user monitoring the message exchange cannot distinguish between a honest smart meter using random values and a malicious smart meter that uses pseudorandom values known to the utility to leak high-frequency data.

If smart meters are assumed to be untrusted, then they cannot be trusted with creating random shares. Therefore, we propose that the user provides the random values used in the computation. We will present such a protocol, under the paradigm "Power to the User" in the next section. The protocol aims at protecting privacy, even if all of the meters are corrupted.

It should be noted that in a setting where smart meters are assumed to be untrusted, their accuracy while recording consumption data is also in question. For example, the smart meter might over- or underreport actual usage during the real-time data aggregation protocol or the billing protocol. This however, can be detected, at least with some effort. For example, old electromechanical meters could be operated in sequence to the smart meter. Readings could then make any deviation of the smart meter apparent.[4] Such an ability of detecting a dishonest smart meter is not possible in the "No-Leakage" setting, since it is not possible to distinguish randomness from pseudorandomness using statistical methods.

3.2.4 User-controlled aggregation protocol

We propose modifications to the protocol, which lead to a new protocol under the "Power to the User" paradigm. The principal concepts are:

1. Introduce the user U_i as a third party to meters and the local station.

2. Allow message exchanges between a smart meter and the local station only via the user.

3. Let the user and the corresponding smart meter each carry out the same computations as in the original protocol, with the user providing the random numbers and encryption parameters.

4. In particular, do so for message encryption: There is a shared key pair (spk, ssk) between the user and the meter. Both smart meter and user have the shared secret key ssk. The key pair is created by the user and then shared with the meter. The encryption scheme used is homomorphic.

5. Let the meter verify the correctness of the user's computations and let it use digital signatures to certify correctness and for integrity protection of the reported shares. This is based on a different key pair (pk, sk) for which the private key is available only to the meter and not the user.

6. Include counters in signatures to protect against replay attacks.

Figure 2 shows the full protocol flow. We do not display the protocol's initialization, where public keys are distributed to all parties and where all parties check whether enough meters participate in the protocol. Some parts of the protocol are simplified to reduce the complexity. For example, we did not include the fact that U_i will send parameters of the underlying encryption scheme to M_i, so that both parties get the same ciphertext.

The counters d_1, d_3 and d_3 are not formally defined. They provide protection against message replay attacks. Their values should not overlap (so that signatures of one step of the protocol can't be used for other steps). If there is a duplicate or missing d value, the protocol should be aborted (we rely on lower-level mechanism to ensure reliable transport). If signature verification fails at any step for any party, the protocol should be aborted as well.

Discussion. Our protocol makes the message flow between the meter and the station transparent to the user. It also aims at eliminating intentional leaks by meters, by letting the user provide random values. Making assertions on the protocol's properties (such as non-leakage) requires rigorous security proofs. Unfortunately, these are not as easy as in the original "No-Leakage" protocol. Introducing a third party and a variety of possible adversaries (meters, users, the local station or any combination) makes a formal analysis more difficult. The complexity of the protocol is also higher, as indicated by the increased number of steps and the signature checks required.

We will therefore only give a short account of what a security proof might show. In contrast to the original protocol, our version takes into account the case, in which all meters

[4] The meter's own consumption has to be taken into account when comparing the meters.

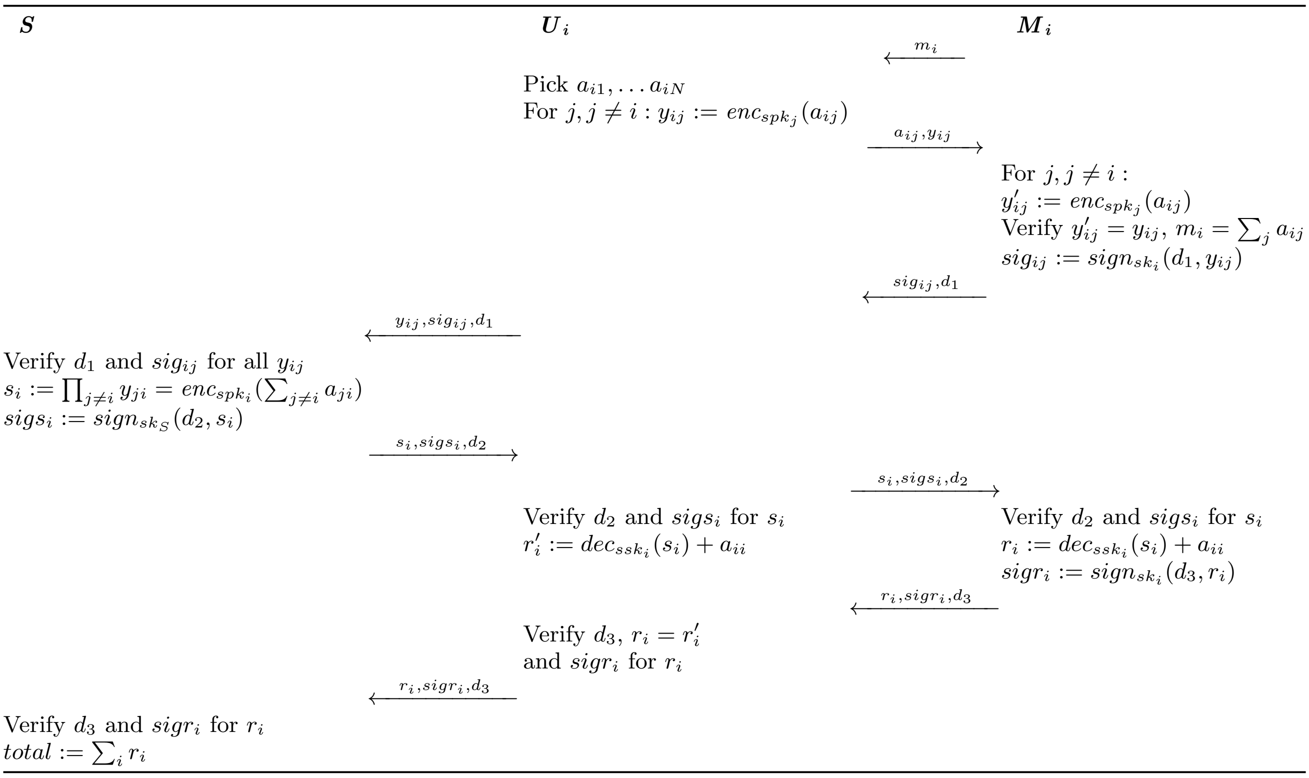
$\boldsymbol{S}$
$\boldsymbol{U}_i$
$\boldsymbol{M}_i$
m_i
Pick $a_{i1}, \dots a_{iN}$
For $j, j \neq i : y_{ij} := enc_{spk_j}(a_{ij})$
a_{ij}, y_{ij}
For $j, j \neq i$:
$y'_{ij} := enc_{spk_j}(a_{ij})$
Verify $y'_{ij} = y_{ij}$, $m_i = \sum_j a_{ij}$
$sig_{ij} := sign_{sk_i}(d_1, y_{ij})$
sig_{ij}, d_1
y_{ij}, sig_{ij}, d_1
Verify d_1 and sig_{ij} for all y_{ij}
$s_i := \prod_{j \neq i} y_{ji} = enc_{spk_i}(\sum_{j \neq i} a_{ji})$
$sigs_i := sign_{sk_S}(d_2, s_i)$
$s_i, sigs_i, d_2$
$s_i, sigs_i, d_2$
Verify d_2 and $sigs_i$ for s_i
$r'_i := dec_{ssk_i}(s_i) + a_{ii}$
Verify d_2 and $sigs_i$ for s_i
$r_i := dec_{ssk_i}(s_i) + a_{ii}$
$sigr_i := sign_{sk_i}(d_3, r_i)$
$r_i, sigr_i, d_3$
Verify d_3, $r_i = r'_i$
and $sigr_i$ for r_i
$r_i, sigr_i, d_3$
Verify d_3 and $sigr_i$ for r_i
$total := \sum_i r_i$

Figure 2: Flow of the proposed user-controlled aggregation protocol

are malicious. We assume that meters record usage data accurately, since this is a property verifiable by the user (using a second meter, as discussed above).[5]

Most important for user privacy is that no consumption information leaks, even if meters and the local station collude. This should hold even if a number of users are part of the collusion. The protocol will not be able to protect the user if he is the only honest party. As with the original protocol, the utility can simply subtract all the other meters' readings from the total consumption to get the uncorrupted user's data.

Users should also not be able to distort the aggregated reading in any way, that is to manipulate the shares such that the total computed by S is not the sum of the individual readings. Integrity protection is introduced for this purpose.

If this protocol indeed has the properties described above, then it is an important improvement over the original protocol described in [4]. Not only is the user able to prevent his meter from leaking information, she can also use the real-time data herself, for her own analysis and purposes.

3.3 Billing

In contrast to real-time aggregation of data, it can be assumed that in billing, each party has a profound interest in manipulating the data, since it affects the final bill. The bill is in its simplest form a sum that is owed by the consumer to the provider. As with the aggregation protocol, it is necessary to provide protection against replay attacks.

In this section, we will sketch a protocol proposed by A. Rial and G. Danezis in [12]. They use homomorphic encryption, zero-knowledge proofs and commitment schemes, to build a secure and privacy-preserving billing system. In conformity with our approach for real-time aggregation systems, this protocol gives the user a great amount of control over the data.

It should be noted that a very simple protocol would create the bill entirely inside the smart meter's trusted computing platform. The bill then is sent over a trusted line to the provider. This has two disadvantages. First of all, it requires that the smart meter carries out all of the computations – which might include billing according to complex tariffs – all in a low-powered, tamper-resistant environment. Secondly, it gives great power to the meter and does not enable the user to verify the data herself, or to do her own computations using the high-frequency data. In contrast, the protocol proposed only requires adding digital signatures to the meter outputs. Everything else for computing the bill is handled by a computer (untrusted by the utility) under the user's control.

[5] Unnoticeably small modifications by the meter of its readings could be used to encode information into the usage reports, if they are not rounded to a lower precision. It might therefore be necessary to include into the protocol preprocessing (quantization) of usage data.

System and design goals. Beside providing security to the provider, the protocol should also provide security to the user. This means, under the assumptions that the individual signed readings are correct, that the provider cannot wrongfully claim to be owed a different sum under a certain tariff than what is actually owed. The user's privacy is the second important design goal, which means that the provider should not learn any high-frequency data. Lastly, meters should not be required to carry out complex calculations, except signing high-frequency reading outputs. As with our real-time data protocol, they only have a data connection to the user, and not the power company or grid operator directly.

The following is the setting which we assume to be in place:

- Meter M outputs the consumption *cons* during a time interval and *other* as information associated with the reading (for example, the time).
- Υ is a publicly known tariff policy, i.e. a pricing function that takes as an input $(cons, other)$ and outputs a *price*. This function is signed by the provider P and sent by it to the user U.
- The tariff policy can be linear (fixed price for each consumption unit during a specific time) or cumulative (price for each consumption unit may depend on how much has been consumed).
- The sum on the bill is calculated by applying the pricing function during each reading and adding up the price of the individual readings. The total on the bill is calculated by the user and sent to the utility, alongside a proof that the total was computed using the meter's readings.
- Each party creates a key pair and distributes its public key to other parties.

Various cryptographic primitives are used in the protocol, which will be presented.

Commitment schemes. A commitment scheme is a cryptographic scheme, where a party can commit to a certain value, without immediately revealing the value. It can later open the commitment by revealing the committed value and the commitment's unique opening. The distinctive feature of commitments is that it's not possible to find a two value/opening pairs that yields the same commitment. After a commitment is opened, all other parties can be assured that the value used for the opening was the value that was committed to.

A classical example for a useful application is the rock, paper, scissors game. Using commitment, each party can commit to a certain value and reveal the commitment's opening along with the committed values only after all commitments are public.

The protocol discussed here uses a homomorphic commitment scheme.

Zero-knowledge proofs. A proof of knowledge of some value that does not reveal the value itself, is called a zero-knowledge proof. It is a protocol between a prover and a verifier. Proofs can either be interactive, requiring message exchanges between both parties (such that the verifier can send challenges to the prover), or non-interactive, consisting of a single message by the prover. The protocol presented here uses non-interactive zero-knowledge proofs.

Protocol flow. For each reading tuple $(cons, other)$, the meter M creates a commitment to *cons* and a commitment *other*. It then creates a digital signature, using its own private key, over $(d, com(cons), com(other))$ (where d is a counter) and sends this signature along with the value d, the commitments and their the openings *cons* and *other* to the user.

The billing protocol is initiated at the end of the billing period to create a final bill from the individual $(cons, other)$ readings and the pricing policy Υ. The entire calculation of the bill, and proof of its correctness, happens outside the smart meter, i.e. on the user's home computer. After the bill has been transmitted to the utility alongside a proof of correctness, the utility verifies the bill and charges the customer with the total on the bill.

For each tuple $(cons, other)$ by the meter, the user creates a commitment to the price $price := \Upsilon(cons, other)$. It then creates a non-interactive zero-knowledge proof π of: (1) knowledge of the signed commitment's opening. (2) knowledge of the opening for the committed price, (3) knowledge of a digital signature by the provider P on $(cons, other, price)$ with which the provider certified that $\Upsilon(cons, other) = price$.

Next, the user uses the homomorphic property of the commitment scheme to aggregate the price commitment, producing a commitment $com(total)$. The user transfers both the commitment and the opening to this total sum $com(total)$ and $(total)$ to the utility. It also includes individual price commitments and the zero-knowledge proofs π. The entire message is signed by the user.

The provider verifies the user's signature on the message, the meters signature on $(d, com(cons), com(other))$ tuples and the proofs π. Afterwards it uses the homomorphic operation on the *price* commitments to get a commitment for the total price. It compares this to $com(total)$ and verifies if the user-reported *total* and opening opens the aggregated *price* commitment. Finally, via the counter d it is verified, if the user did include exactly the readings for the billing period.

The protocol requires that the user can be asked by the provider to give up some $(cons, other)$ tuples, by providing the openings to the committed values.

Outlook. The construction used by Rial and Danezis is too complex to fully explain in this paper. The same holds for the consideration of different tariff structures and the security proofs. However, the sketched protocol flow above outlines for which part of the protocol the cryptographic primitives presented above are required. It also shows, that while difficult, creating privacy-preserving billing protocols that allow for tariff computations outside smart meters is feasible.

4. CONCLUSIONS

In this paper, we have presented and discussed several solutions to the challenges brought up by smart meter technology. While current smart meters deployments largely ignore risks for the user' privacy, the results above show that these threats are substantial. We have argued that the problems are not of the form of an unresolvable conflict between privacy and the benefits that smart meters are claimed to introduce. To the contrary, over the past years a number of researchers have presented sophisticated approaches to deal with problematic high-frequency data. Even though providers are reluctant to share their requirements for real-time data collection, researchers are able to make educated guesses about which data is actually needed.

For the case of real-time data collection, we have seen three different proposals. While preprocessing helps to reduce the severity of the issues at hand, it does not fundamentally resolve them. As for the second approach, we argued that a protocol requiring that providers do not collude in order for privacy protections to be effective, provides little protection when adveraries are government agencies. The "No-Leakage" protocol presented is a promising approach to protect privacy while enabling real-time data collection. Instead of distributing the control over multiple providers, the control is given to the smart meters. Thus, in order for information to leak, almost all meters have to be corrupted.

This seems like a satisfying result, until one considers the influence that providers exert over the smart meter deployment. Attacks on privacy by providers who roll out corrupted (i.e. backdoored) smart meters to their customers form an adversarial model overlooked in the protocol's design. In such a scenario, the data protection mechanism fails. Worse yet, the user is unable to detect attacks. Without inspection of the software running on the tamper-resistant device, the user cannot decide whether the smart meter leaks data or honestly follows the protocol. This holds even if she is able to read all messages exchanged between the smart meter and the provider.

To protect the user against such attacks, we proposed a protocol similar to "No-Leakage", with the difference, that the user is in control. She provides the randomness used in the encryption key and the random shares. To prevent misuse, the smart meter verifies the correctness of the user's calculations and protects their integrity with a digital signature.

The protocol aims at ensuring privacy even if smart meters behave maliciously. However, more research is required to subject the protocol to a thourough security analysis and to make formally proven statements on its properties.

A further topic of research is the distribution of public keys to all parties. A public-key infrastructure with certificate authorities scales well, but has problems: If certificate authorities are run or controlled by providers, then trust cannot be established, since they are assumed to be possible adversaries. The users for one substation live in close proximity, such that alternatively a personal key exchange and a web-of-trust model might be feasible.

Lastly, we looked at the billing use case. There, security and privacy were in conflict in particular, since each party has an interest in manipulation of the bill. While it was only possible to sketch one approach in the restricted scope of this paper, it became apparent that complex tariffs can be implemented in a privacy-preserving manner.

The architectures both of the billing protocol and our aggregation protocol follow the "Power to the User" paradigm. We argued that in order for customers to have confidence in the protection of their privacy, it does not suffice to transfer control from the providers to the meters. The user must not only be able to read the message exchange, but act as an active participant in the protocols. Although further research is required to refine, evaluate and formally analyze the protocols presented in this paper, they are promising approaches for secure and privacy-preserving smart meters systems.

5. REFERENCES

[1] Directive 2009/72/EC of the European Parliament and of the Council, 13 July 2009.

[2] Smart Grid Projects Outlook, 2014. `http://ses.jrc.ec.europa.eu/smart-grids-observatory`

[3] C. Barcellona, P. Cassara, G. Di Bella, J. Golic, and I. Tinnirello. Multi-party metering: An architecture for privacy-preserving profiling schemes. In *Sustainable Internet and ICT for Sustainability (SustainIT), 2013*, pages 1–6, Oct 2013.

[4] F. Garcia and B. Jacobs. Privacy-friendly energy-metering via homomorphic encryption. In J. Cuellar, J. Lopez, G. Barthe, and A. Pretschner, editors, *Security and Trust Management*, volume 6710 of *Lecture Notes in Computer Science*, pages 226–238. Springer Berlin Heidelberg, 2011.

[5] J.-H. Hoepman. Privacy design strategies. *CoRR*, abs/1210.6621, 2012.

[6] P. Kumari, F. Kelbert, and E. Pretschner. Data protection in heterogeneous distributed systems: A smart meter example, October 2011.

[7] K. Kursawe, G. Danezis, and M. Kohlweiss. Privacy-friendly aggregation for the smart-grid. In *Proceedings of the 11th International Conference on Privacy Enhancing Technologies*, PETS'11, pages 175–191, Berlin, Heidelberg, 2011. Springer-Verlag.

[8] C. McKerracher and J. Torriti. Energy consumption feedback in perspective: integrating australian data to meta-analyses on in home displays. *Energy Efficiency*, 6(2):387–405, May 2013.

[9] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin. Private memoirs of a smart meter. In *Proceedings of the 2Nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, BuildSys '10, pages 61–66, New York, NY, USA, 2010. ACM.

[10] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In J. Stern, editor, *Advances in Cryptology – EUROCRYPT 1999*,

volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer Berlin Heidelberg, 1999.

[11] A. Reinhardt, F. Englert, and D. Christin. Enhancing user privacy by preprocessing distributed smart meter data. In *Sustainable Internet and ICT for Sustainability (SustainIT), 2013*, pages 1–7, Oct 2013.

[12] A. Rial and G. Danezis. Privacy-preserving smart metering. In *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society*, WPES '11, pages 49–60, New York, NY, USA, 2011. ACM.

[13] L. Sankar, S. Kar, R. Tandon, and H. V. Poor. Competitive privacy in the smart grid: An information-theoretic approach. *CoRR*, abs/1108.2237, 2011.

[14] C. Thoma, T. Cui, and F. Franchetti. Secure multiparty computation based privacy preserving smart metering system. In *North American Power Symposium (NAPS), 2012*, pages 1–6, Sept 2012.